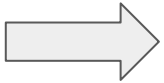
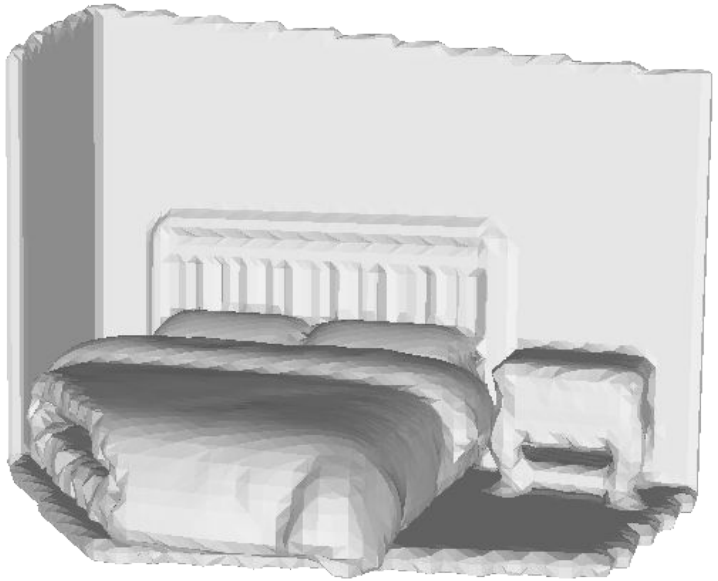
3D Reconstruction from a Single RGB Image

Alexander Sheldrick & Moustafa Elsharkawy

Supervised by Yawar Siddiqui

Task





Single RGB image with camera intrinsics and occupancy points sampled from the GT mesh for training

Reconstructed Geometry

Fu et. al: “3D-FRONT: 3D Furnished Rooms with layOuts and semaNTics”, 2020

Related Work

IF-Nets

- IF-Nets works with a variety of 3D inputs: low/high-resolution voxel-grids, sparse/dense point-clouds, complete or incomplete.

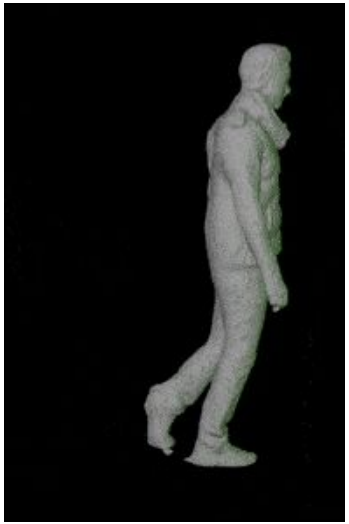
Output 3D Repr.	Continuous Output	Multiple topologies	Sparse Input	Dense Input	Arti-culated
Voxels	✗	✓	✓	✗	✓
Points	✗	✓	✓	✗	✓
Meshes	✗	✗	✓	✗	✓
Implicit*	✓	✓	✓	✗	✗
IF-Nets	✓	✓	✓	✓	✓

Chibane et. al: “Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion”, 2020

Related Work

IF-Nets

- Focuses on 3D surface reconstruction and shape completion from a variety of 3D inputs.
- Extracts deep features from the input at continuous point locations. Based only on these features a decoder decides whether the point p lies inside or outside the surface.
- This improves on prior work by classifying points based on local and global shape features, instead of point coordinates.

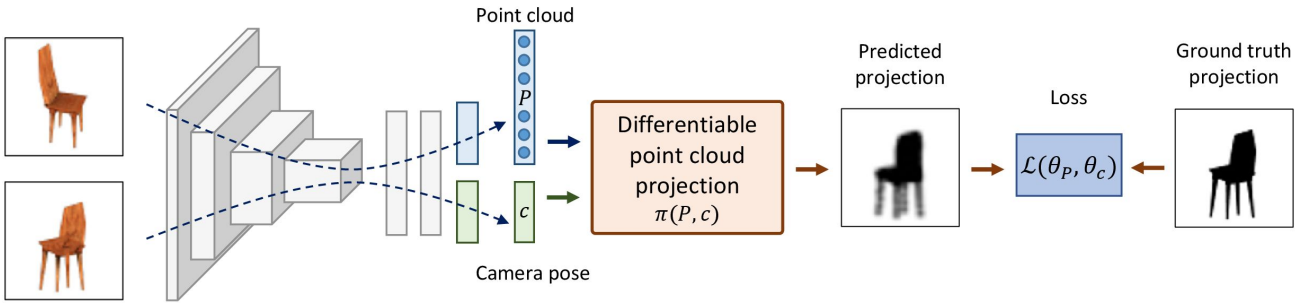


Chibane et. al: “Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion”, 2020

Related Work

Unsupervised Learning of Shape and Pose with Differentiable Point Clouds

- Predicts both the shape and the pose from a single image by minimizing the reprojection error.
- Converts the point cloud representation to a discretized occupancy grid in a differentiable manner.



Eldar Insafutdinov and Alexey Dosovitskiy: "Unsupervised Learning of Shape and Pose with Differentiable Point Clouds", 2018

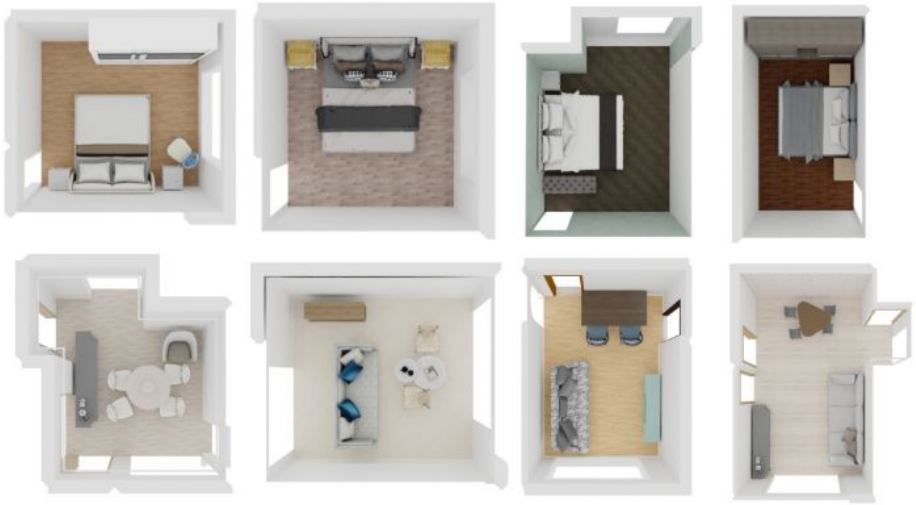
Our Approach & Contributions

- Explore the use of IF-Nets for the task of 3D reconstruction from images.
- Explore the ability for IF-Nets to work on complex scenes instead of shapes.

Dataset

3D-FRONT:

- Offers a variety of synthetic indoor scenes.
- All scene details can be extracted: ground truth depth maps and distance fields.

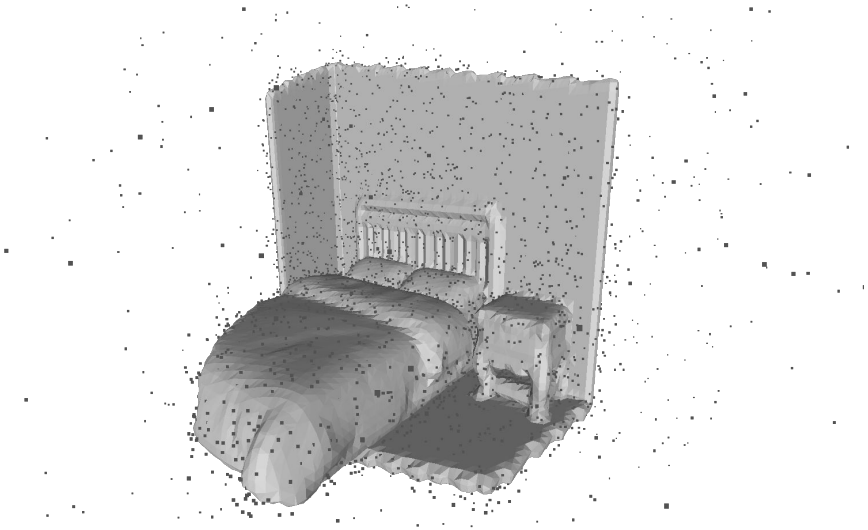


Fu et al. “3D-FRONT: 3D Furnished Rooms with layOuts and semaNTics”. 2020

Pipeline

Mesh sampling:

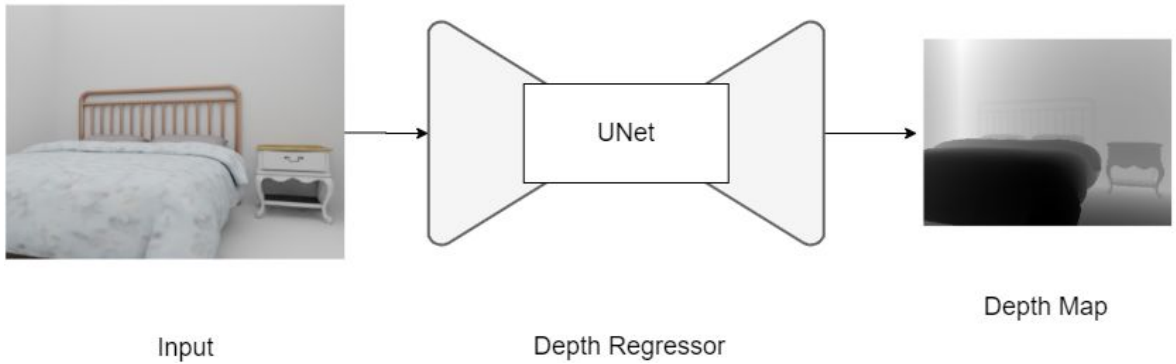
- Transform the ground truth distance field into a mesh (marching cubes).
- Make the mesh watertight.
- From the mesh, we randomly sample points with a predefined variance.
- Compute the ground truth occupancy for each sampled point.
- These points are only used for training.



Pipeline

U-Net

- U-Net feature concatenation allows for a more precise depth estimation.
- U-Net introduces restrictions on input size.

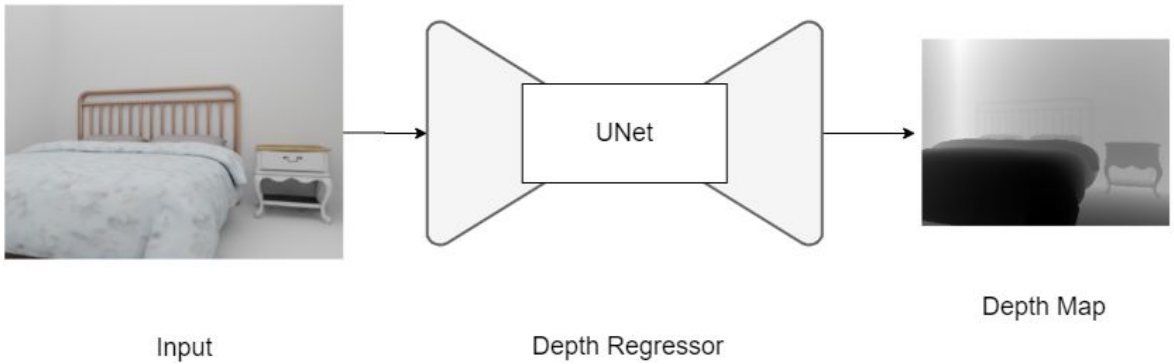


O. Ronneberger et al. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. MICCAI 2015

Pipeline

U-Net

- Square padding the image with zeros and resizing the input.
- The output depth is resized back and the padding is discarded.
- This is followed by a sigmoid layer and a renormalization step.

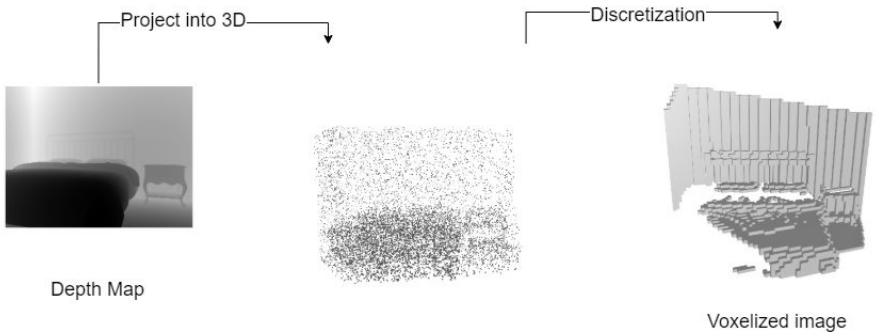


O. Ronneberger et al. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. MICCAI 2015

Pipeline

2D-to-3D Projection

- Project the depth map into 3D point cloud.
- Then discretize the point cloud into an occupancy grid.
- For end-to-end pipeline training, discretization needs to be differentiable.

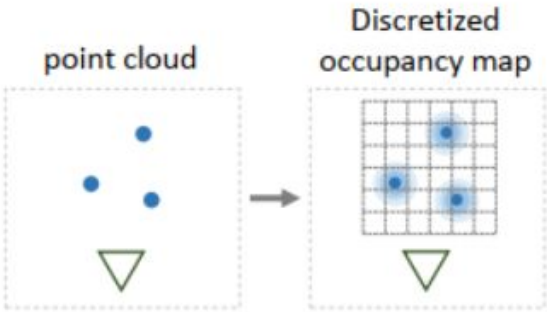


Eldar Insafutdinov and Alexey Dosovitskiy: “Unsupervised Learning of Shape and Pose with Differentiable Point Clouds”, 2018

Pipeline

Differentiable voxelization:

- Smooth the density of each point with a Gaussian distribution.
- The resulting voxelization is then the clipped sum of smoothed points evaluated at the voxel grid points.

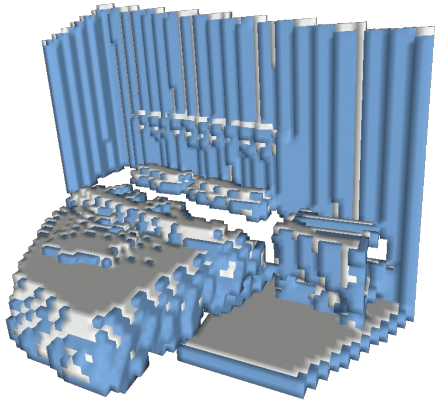


Eldar Insafutdinov and Alexey Dosovitskiy: “Unsupervised Learning of Shape and Pose with Differentiable Point Clouds”, 2018

Pipeline

Differentiable voxelization implementation:

- First, put all the points on a grid with trilinear interpolation.
- Second, apply 3D convolution over the grid with gaussian kernels (learnable sigmas).
- Gradients can flow enabling end-to-end training.



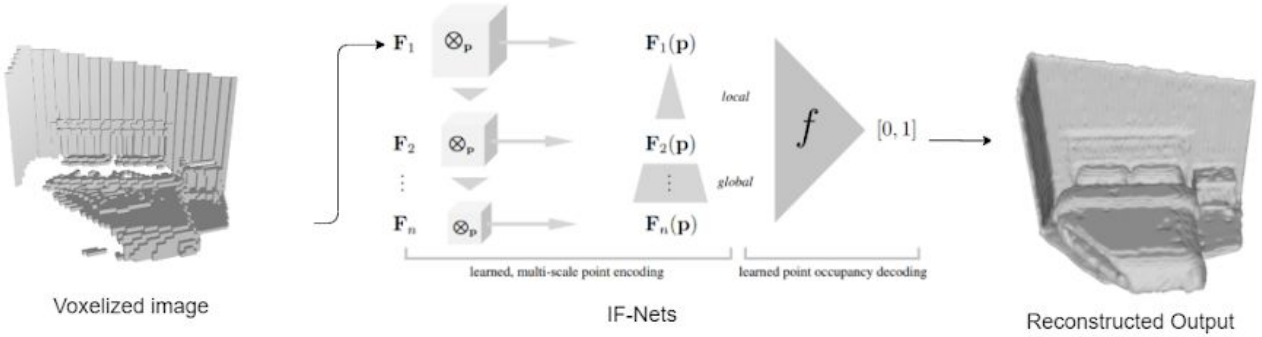
Voxelized depth
Differentiable voxelization

Eldar Insafutdinov and Alexey Dosovitskiy: “Unsupervised Learning of Shape and Pose with Differentiable Point Clouds”, 2018

Pipeline

If-Net

- Takes as input the voxelized image and a query point to determine its occupancy.
- Using 3D convolutions, we obtain a 3D grid of features at multiple scales.

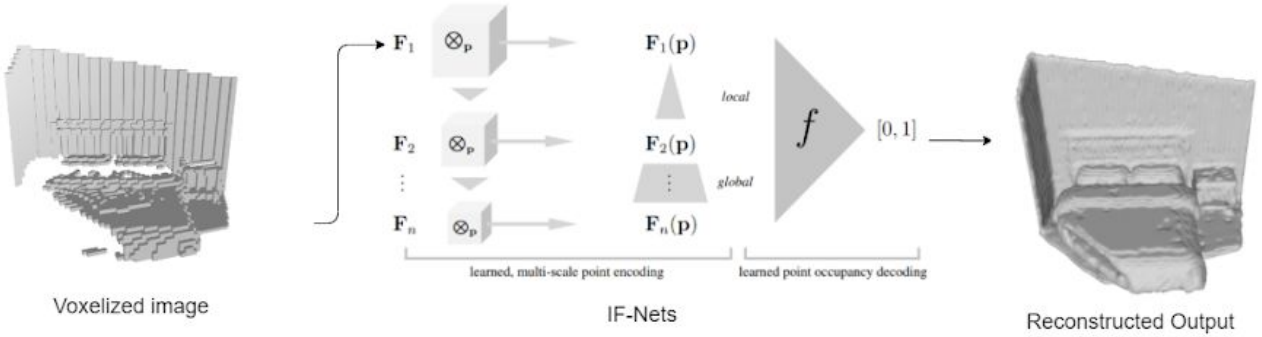


Chibane et. al: "Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion", 2020

Pipeline

If-Net

- Extract features at the location of a query point p itself and additionally at surrounding points in a distance d along the Cartesian axes.
- These features are fed to a fully connected neural network, to predict if the point p lies inside or outside the shape.

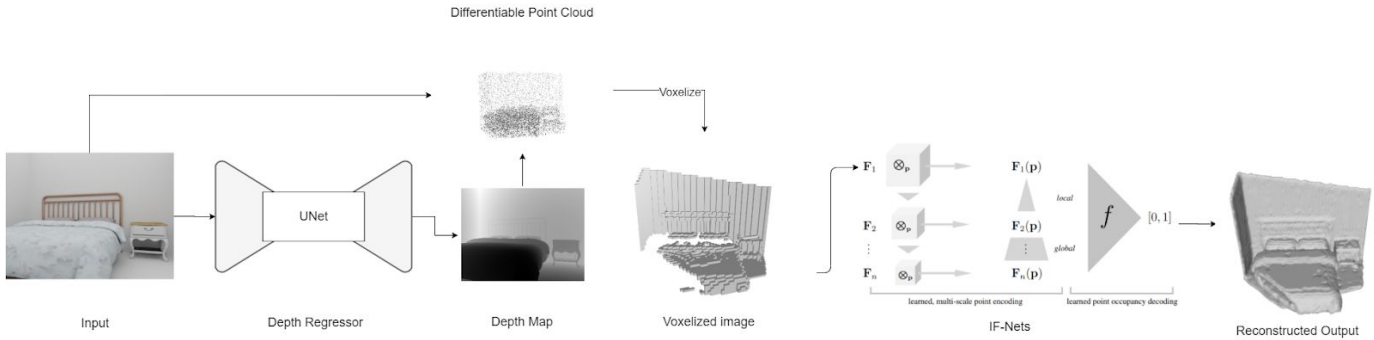


Chibane et. al: "Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion", 2020

Pipeline

Training vs. Inference

- During training: points from the sampling and the point cloud are combined.
- During Inference: Evaluate the full network on points on a grid of the desired resolution to get the occupancy grid.



Chibane et. al: "Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion", 2020

Experimentation

Bit precision

- Mixed vs. full precision training.
- Full precision training converged more quickly to lower validation losses.
- Due to a known PyTorch Bug, ’torch.nn.functional.grid sample’ (which IF-Net uses frequently) runs roughly 10x faster in 32bit than in 16bit.

Experimentation

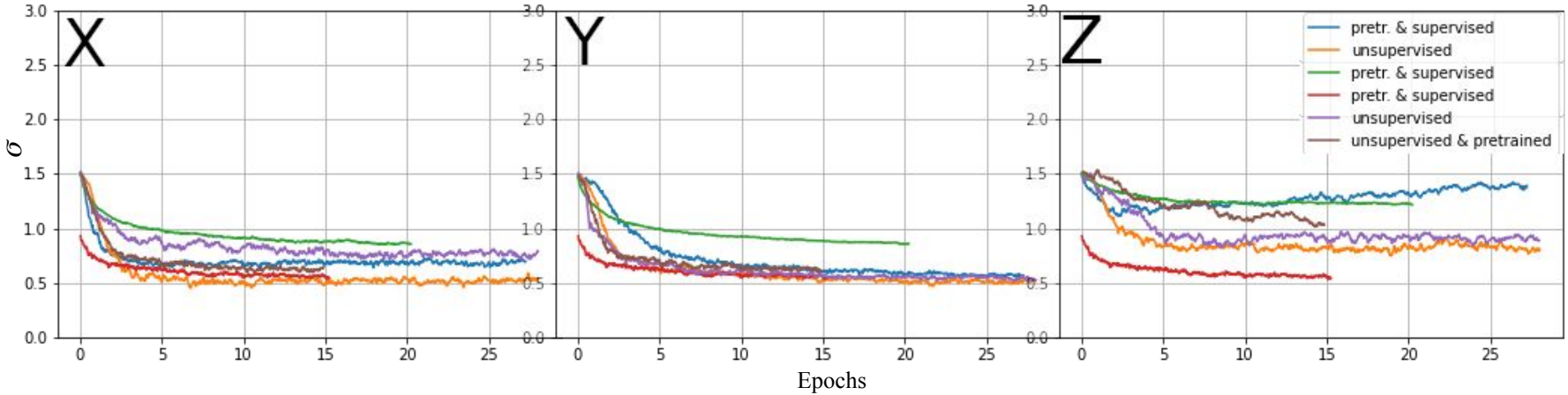
Training speed

- Training was very slow due to limited resources and computation power.
- We needed to speed up the training without compromising the results.
- Therefore, we implemented scaling of the occupancy grid that is fed as input to the IF-Net.
- Additionally, we only used a subset of the point cloud which greatly reduced our gpu memory usage.
- Speed up by 10 times which allowed us to experiment faster for our results.

Experimentation

Smoothing covariance and kernel size

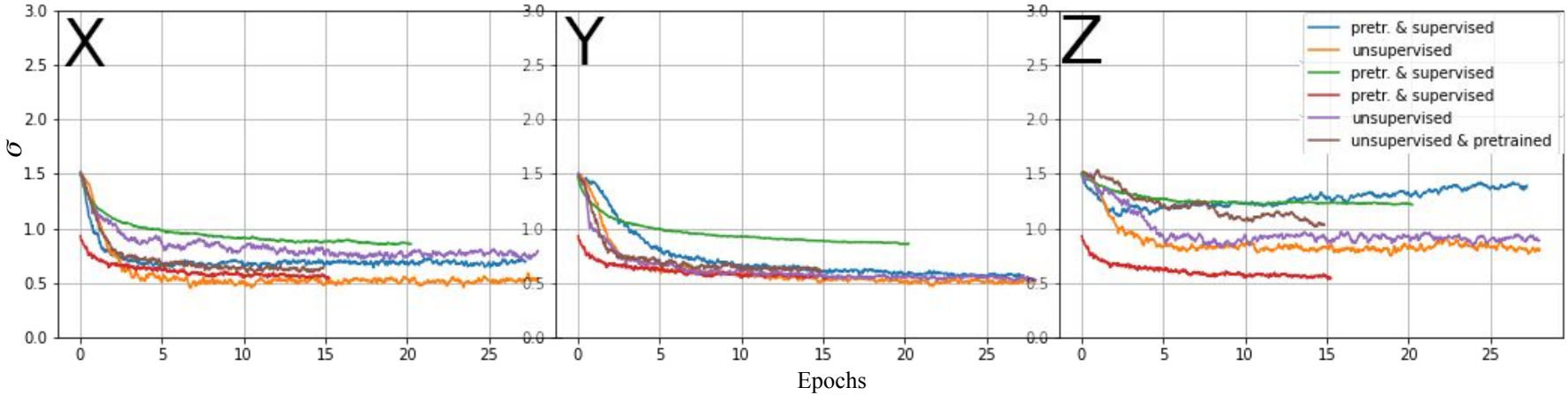
- Using an isotropic covariance, the network learned sigma converged to just above 0.5 regardless of kernel-size
- we interpret as the uncertainty of point positions of the order of half a voxel.



Experimentation

Smoothing covariance and kernel size

- After splitting the standard deviations per dimension, model now learns different parameters.
- Highest source of uncertainty for model is depth-prediction (regardless of sup.).

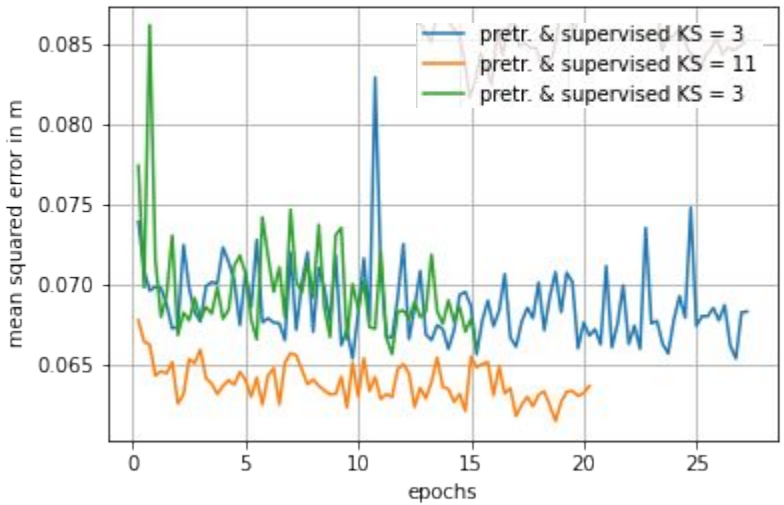


Depth regression validation losses

Validation losses with depth-supervision

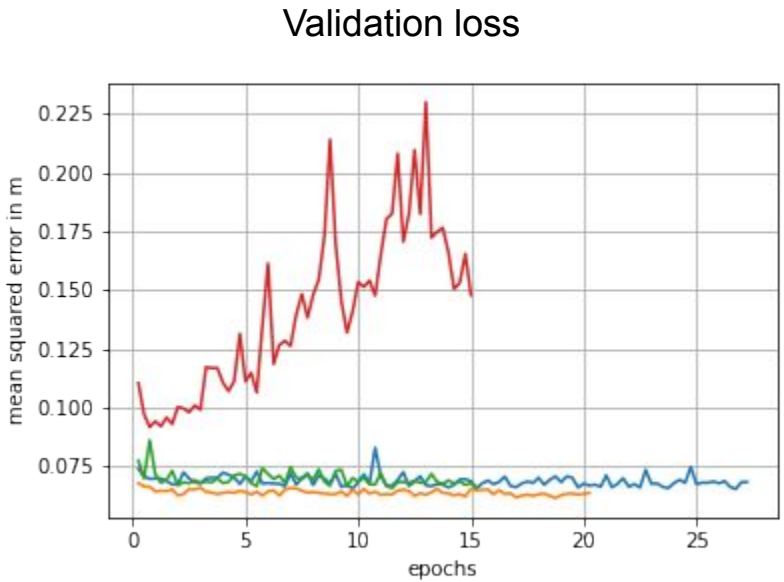
MSE losses roughly on the order of 1 Voxel

Validation loss



Depth regression validation losses

Training a new model with pre-trained depth-regressor slowly diverges per pixel loss



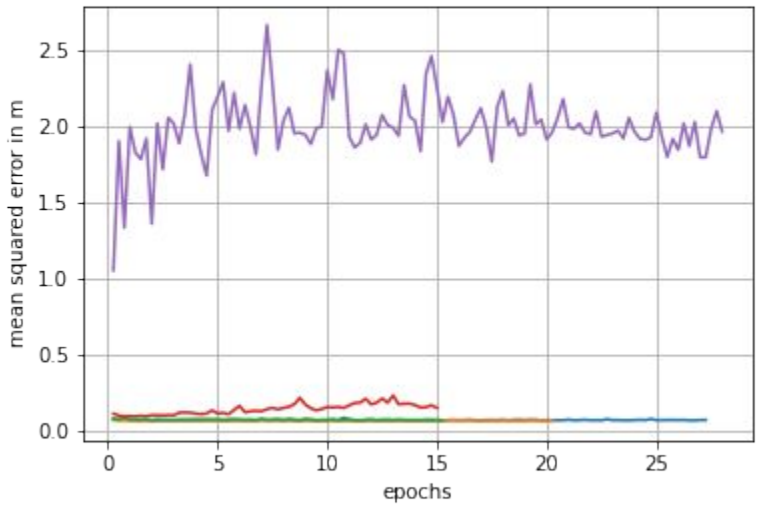
Depth regression validation losses

Initializing with an untrained depth-regressor gives widely diverging depth losses

Results are still good, so probably not a depth-regressor but feature encoder now.

Breaks interpretability as voxelgrid and objects don’t look much alike

Validation loss



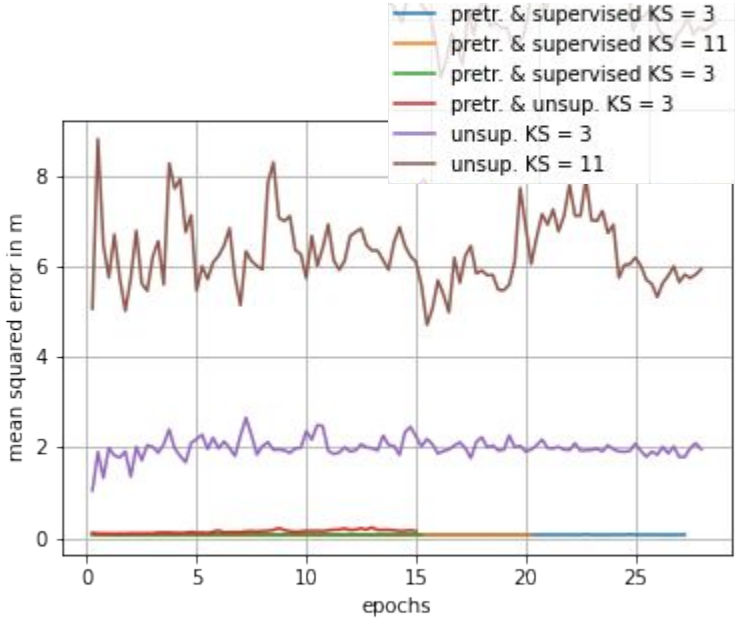
Depth regression validation losses

Initializing with an untrained depth-regressor gives widely diverging depth losses

Results are still good, so probably not a depth-regressor but feature encoder now.

Breaks interpretability as voxelgrid and objects don’t look much alike

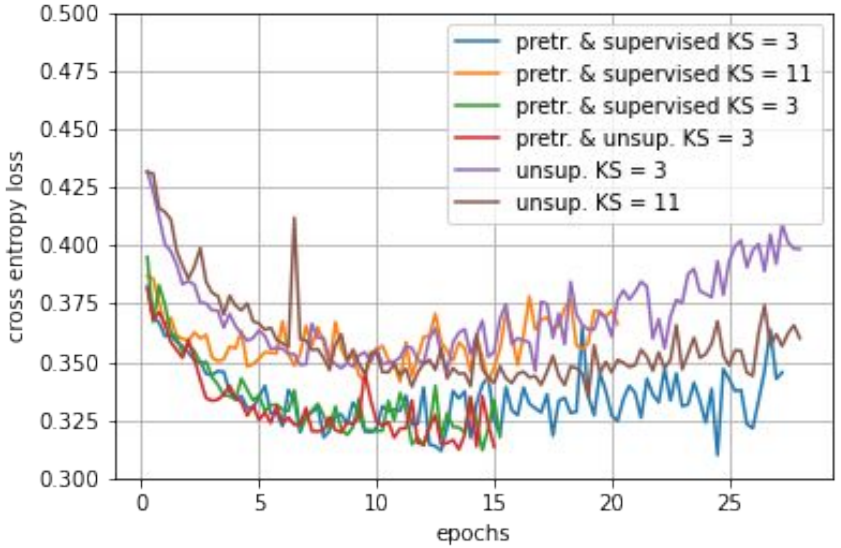
More experiments required how kernel size affects test results



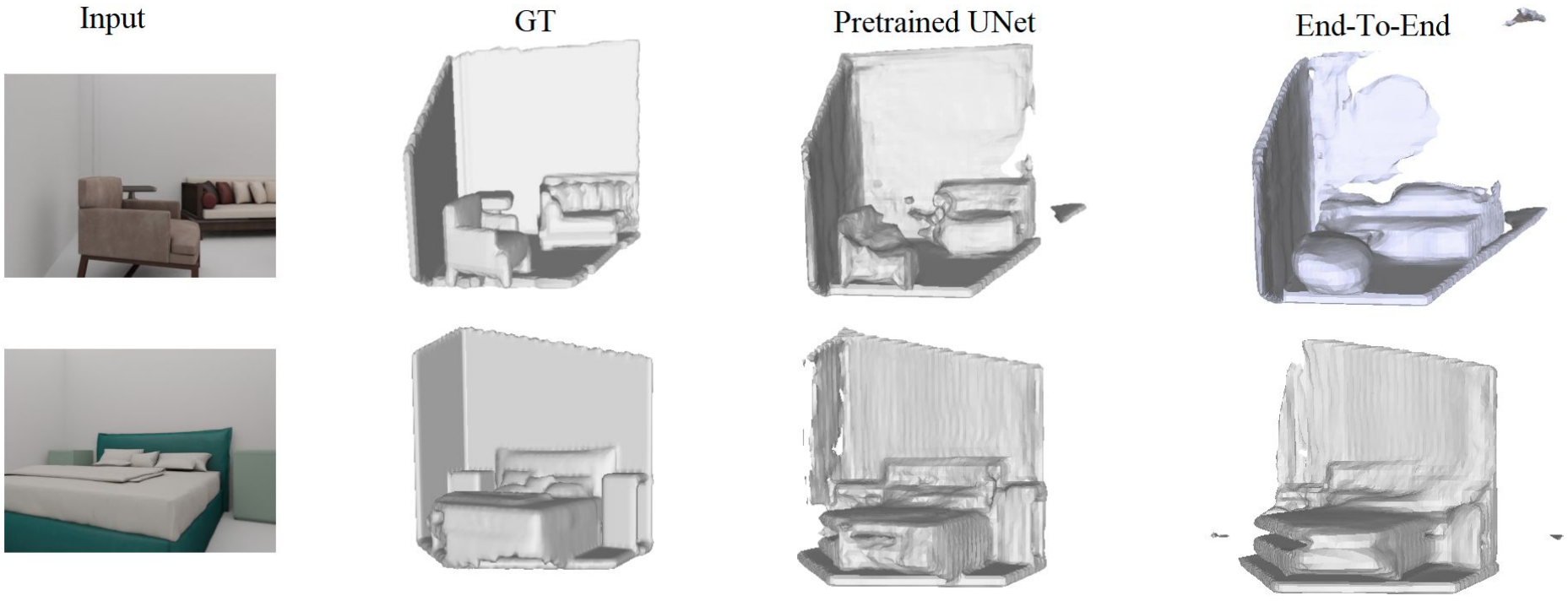
Binary cross entropy validation loss

Overfit after ~15 Epochs (epoch ~25k samples / 30min)

Models perform within 5% (IoU) of another



Qualitative Results



Quantitative Results

	↑ IoU	↓ Chamfer-L2	↑ Normal-Consistency
IF-Net*	**0.73**	**0.00002**	**0.91**
Viability**	0.48	0.0007	0.82
Pretr. + sup.	0.43	0.0065	0.82
End-to-End	0.40	0.0094	0.77

Meshes are normalized and centered [-0.5, 0.5] prior to evaluation
*Author's reported results on different data-set (ShapeNet)
**Trained on smaller, downsized data-set, voxelizing GT depth

Conclusion

- Due to time and computational resource limits, experiments are not fully optimized - motivation for further research.
- IF-Net works well for the task of the task of 3D reconstruction.
- The estimated depth is the major shortcoming.

Thank you

Appendix

Quantitative Results

- IoU

$$\mathrm{IoU}(\mathcal{M}_{\mathrm{pred}}, \mathcal{M}_{\mathrm{GT}}) \equiv \frac{|\mathcal{M}_{\mathrm{pred}} \cap \mathcal{M}_{\mathrm{GT}}|}{|\mathcal{M}_{\mathrm{pred}} \cup \mathcal{M}_{\mathrm{GT}}|}$$

- Normal consistency

~ integral over dot product of unit norm vectors of mesh surfaces

- Chamfer-L2

~ squared distance between sampled points and corresponding mesh surface

Mescheder et. al: “Occupancy Networks: Learning 3D Reconstruction in Function Space”, 2019